

C Yuyan Chengxu Sheji



高等学校“十五”规划教材

C语言程序设计

主 编 贾宗璞 许合利

中国矿业大学出版社



高等学校“十五”规划教材

C 语言程序设计

主 编 贾宗璞 许合利

中国矿业大学出版社

内 容 简 介

本书是高等院校计算机基础教育用教材,全书共分14章,主要内容包括C语言概述,基本数据类型、运算符与表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,编译预处理,结构体、共用体及枚举类型,文件,C++基础,集成开发环境及程序测试与调试,实验等,各章后并附有习题。书后附有常用字符与ASCII代码对照表、C语言中的关键字、运算符优先级和结合方向、Turbo C常用库函数。

本书内容丰富、新颖,图文并茂,通俗易懂,实用性强,可作为高等学校非计算机专业的计算机基础课教材,也可作为应用计算机人员的自学参考书。

图书在版编目(CIP)数据

C语言程序设计/贾宗璞,许合利主编. —徐州:中国矿业大学出版社, 2007. 1

ISBN 978-7-81107-501-4

I. C… II. ①贾…②许… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第020637号

书 名 C语言程序设计

主 编 贾宗璞 许合利

责任编辑 孙树朴

责任校对 王春风

出版发行 中国矿业大学出版社

(江苏省徐州市中国矿业大学内 邮编:221008)

网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com

印 刷 北京兆成印刷有限责任公司

经 销 新华书店

开 本 787×1092 1/16

印 张 24.25

字 数 449千字

版次印次 2007年2月第1版 2007年2月第1次印刷

印 数 1~11000册

定 价 29.00元

(若图书出现印装质量问题,本社负责调换)

前 言

C 语言是国际上广泛流行的一种面向过程的计算机高级语言,其历史悠久,发展相当迅速。后来,发展起来的 C++、Java 等语言,无不是在其基础上进行扩充的。C 语言与其他高级语言相比,形式简洁,数据类型丰富,表达能力强,运算丰富,程序设计灵活;可读性和可移植性好,目标程序效率高,既具有低级语言的特点,又具有完善的模块化结构,体现了结构化程序设计的思想,适合于培养良好的编程风格和优秀的程序设计技术的训练。它是继 PASCAL 语言之后的又一门优秀的课程教学语言,并且是教学需要与实际应用相结合的一门语言。C 语言具有很强的处理功能,不仅用于开发系统软件,也可用于开发应用软件。

学习 C 语言,起初会觉得要记的东西太多,这是由于它太灵活了。但是,学到一定程度,就会尝到甜头,就会体会出 C 语言的特色。C 语言中的指针是一个核心,是今后开发工作中的得力助手,因为在使用 C/C++ 的实际工作中指针无处不在,很多参数就完全是指针化的。虽然 Java 从安全性方面考虑摒弃了指针,但从性能上来说,却得不偿失。要学好 C 语言,就要透彻理解概念,辅之以大量编程训练和上机实验。只靠看书学不好 C 语言,要积极实践,善于思考,结合具体的项目(哪怕是很小的项目)学用相长。坚持下去,就会成功。

本书是根据教育部高等学校计算机科学与技术教学指导委员会编制的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》编写的,属于较高要求的 C 语言版本。全书共分 14 章,第 1 章介绍了 C 语言的基本知识、算法及程序设计的一般方法;第 2 章介绍了 C 语言的基本数据类型、常量和变量以及表达式;第 3 章、第 4 章、第 5 章介绍了 C 语言进行结构化程序设计的基本方法,包括结构化程序的顺序结构、选择结构、循环结构及其设计方法;第 6 章介绍了数组;第 7 章介绍了函数;第 8 章介绍了编译预处理;第 9 章充分阐述了 C 语言的指针;第 10 章对结构体、共用体、枚举作了较详细的介绍;第 11 章对 C 语言文件操作作了较详细的阐述;第 12 章对 C++ 的基本特性、基本概念以及面向对象程序设计的基本方法进行了初步阐述;第 13 章介绍了 Turbo C 2.0 集成开发环境的使用、上机实验的指导思想和要求、程序调试与

测试方法、常见出错信息与分析;第14章介绍了配合教学的实验,以及各实验项目的题目、要求和步骤。另外,每章都配有本章重点、难点和小结,并附有大量的习题,在书末还有附录来指导大家的学习。

作者多年从事C语言及计算机相关课程的教学实践,成稿前作者曾多次编写讲义、教辅资料、习题集。本书是在此基础上,经过认真讨论,集思广益,精心整理、编写而成的。本书在内容选取上既注意了先进性、科学性和系统性,又兼顾了实用性;在文字叙述上力求做到深入浅出,通俗易懂,便于自学;同时用大量的典型实例化解各章的难点,充分展示了计算机解决问题的思想和方法,突出了程序设计的基本方法的阐述。另外,作者多年来一直参与全国计算机等级考试的组织、辅导工作,对全国计算机等级考试的大纲有透彻的理解,所以本书将大纲中二级C语言的要求贯穿其中。每章习题的格式糅合了等级考试的笔试和机试两部分。因此,本书除了可以作为普通高校各专业学生的教材外,还可以作为参加全国计算机等级考试的参考用书,也可以供学习C语言的科技人员使用。

本书的编写人员均为河南理工大学多年从事计算机教学的教师。主编为贾宗璞、许合利。具体编写分工为:贾宗璞编写第1章、第10章;许合利编写第2章、第9章;杨波编写第3章、第7章;王永茂编写第4章、第6章;郭海儒编写第8章、第12章;徐光明编写第5章、第11章;崔志恒编写第13章及附录;王瑞、董玉杰编写第14章。在本书的编写过程中,得到了河南理工大学领导和教务处的的大力支持,在此表示衷心感谢。

由于作者水平有限,书中缺点、错误在所难免,敬请读者批评指正。

编 者

2006年12月

目 录

第 1 章 C 语言概述	(1)
1.1 C 语言的发展及特点	(1)
1.2 C 语言程序的基本结构	(4)
1.3 算法及其描述	(7)
1.4 程序设计方法	(14)
本章小结	(16)
习题	(17)
第 2 章 基本数据类型、运算符与表达式	(19)
2.1 C 语言的数据类型	(19)
2.2 常量和变量	(21)
2.3 整型数据	(23)
2.4 实型数据	(26)
2.5 字符型数据	(28)
2.6 算术运算符与算术表达式	(31)
2.7 赋值运算符与赋值表达式	(34)
2.8 逗号运算符与逗号表达式	(35)
2.9 位运算符	(36)
2.10 数据类型转换与计算类型长度运算符	(38)
本章小结	(41)
习题	(42)
第 3 章 顺序结构程序设计	(46)
3.1 C 语言语句概述	(46)
3.2 字符数据的输入输出	(48)
3.3 格式化输入输出函数	(49)
3.4 顺序结构程序设计举例	(58)
本章小结	(59)
习题	(60)

第 4 章 选择结构程序设计	(64)
4.1 关系运算符和关系表达式	(64)
4.2 逻辑运算符和逻辑表达式	(66)
4.3 if 语句及其构成的选择结构	(68)
4.4 switch 语句及其构成的选择结构	(75)
4.5 选择结构程序设计举例	(77)
本章小结	(80)
习题	(80)
第 5 章 循环结构程序设计	(87)
5.1 概述	(87)
5.2 while 语句	(88)
5.3 do-while 语句	(90)
5.4 for 语句	(92)
5.5 break 语句和 continue 语句	(94)
5.6 循环的嵌套	(96)
5.7 循环结构程序设计举例	(97)
本章小结	(100)
习题	(101)
第 6 章 数组	(109)
6.1 一维数组	(109)
6.2 二维数组与多维数组	(115)
6.3 字符数组与字符串	(119)
本章小结	(126)
习题	(127)
第 7 章 函数	(132)
7.1 模块化程序设计	(132)
7.2 函数的定义	(135)
7.3 函数调用	(137)
7.4 函数的递归调用	(141)
7.5 数组作为函数参数	(144)
7.6 变量的作用域	(147)
7.7 变量的存储类别	(150)
7.8 内部函数和外部函数	(155)

本章小结	(156)
习题	(157)
第 8 章 编译预处理	(163)
8.1 宏定义	(163)
8.2 条件编译	(171)
8.3 文件包含	(174)
本章小结	(176)
习题	(177)
第 9 章 指针	(179)
9.1 地址和指针的概念	(180)
9.2 指针变量	(181)
9.3 指针与数组	(184)
9.4 指针与字符串	(192)
9.5 指针与函数	(195)
9.6 指针数组和多级指针	(204)
本章小结	(210)
习题	(211)
第 10 章 结构体、共用体及枚举类型	(219)
10.1 结构体变量的定义	(219)
10.2 结构体变量的引用和初始化	(222)
10.3 结构体数组	(223)
10.4 结构体指针变量	(226)
10.5 结构体与函数	(229)
10.6 位段结构体	(232)
10.7 链表	(235)
10.8 共用体	(245)
10.9 枚举类型	(248)
10.10 用 typedef 定义类型	(250)
本章小结	(251)
习题	(252)
第 11 章 文件	(258)
11.1 文件概述	(258)
11.2 文件类型指针和文件位置指针	(260)

11.3	文件的打开和关闭	(261)
11.4	文件的读写	(264)
11.5	文件的定位和出错检测	(270)
	本章小结	(272)
	习题	(272)
第 12 章	C++ 基础	(279)
12.1	概述	(279)
12.2	C++ 对 C 的扩充	(282)
12.3	类与对象	(291)
12.4	继承与派生	(298)
12.5	多态性与虚函数	(305)
	本章小结	(309)
	习题	(310)
第 13 章	集成开发环境及程序测试与调试	(313)
13.1	Turbo C 2.0 集成开发环境的使用	(313)
13.2	Borland C++ 3.1 的使用	(325)
13.3	程序测试与调试	(327)
13.4	上机实验总目的和要求	(333)
13.5	常见编译出错信息	(334)
第 14 章	实验	(341)
14.1	实验一 基本数据类型、运算符与表达式	(341)
14.2	实验二 顺序和选择程序设计	(344)
14.3	实验三 循环程序设计	(347)
14.4	实验四 数组	(349)
14.5	实验五 函数(1)	(350)
14.6	实验六 函数(2)与编译预处理	(353)
14.7	实验七 指针	(354)
14.8	实验八 结构体、共用体与枚举类型	(358)
14.9	实验九 文件	(361)
14.10	实验十 C++ 基础	(363)
14.11	实验十一 综合程序设计	(364)

附录 I	常用字符与 ASCII 代码对照表	(367)
附录 II	C 语言中的关键字	(368)
附录 III	运算符优先级和结合方向	(369)
附录 IV	Turbo C 常用库函数	(371)
主要参考文献		(378)

第1章 C语言概述

本章重点

※ C语言的特点

※ C语言程序的基本结构

※ 算法及其描述方法

※ 结构化程序设计方法

本章难点

※ C语言与其他高级语言的区别

※ 算法的流程图、N-S图描述方法

计算机本身是无生命的机器,要使之为人完成各种各样的工作,就必须让它执行预先设计的相应程序。这些程序都是用程序设计语言编制出来的。在众多的程序设计语言中,C语言有其独特优点,深受软件工作者欢迎。本章主要从程序设计的角度,介绍C语言的发展及特点,描述C语言程序的基本结构、算法以及程序设计方法等。

1.1 C语言的发展及特点

1.1.1 C语言的发展

在C语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件,所以其可读性和可移植性都很差;一般的高级语言,难以像汇编语言一样实现对计算机硬件直接操作。于是人们盼望能有一种兼有汇编语言和高级语言特性的新语言。

在这种背景下,C语言于20世纪70年代初问世。当时,主要是用于UNIX系统的开发,原来的UNIX操作系统是1969年由美国贝尔实验室的K. Thompson和

D. M. Ritchie 用汇编语言开发而成的。后改用 B 语言实现,但 B 语言过于简单,功能有限。为了更好地描述和实现 UNIX 操作系统,1972 年至 1973 年间,贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了最初的 C 语言。几经修改后,1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时,由 B. W. Kernighan 和 D. M. Ritchie 合著了著名的《The C Programming Language》一书,通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言,后来由美国国家标准化协会 ANSI(American National Standards Institute)在此基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。

随着人们对 C 语言的强大功能和各方面的优点的逐步了解,到了 20 世纪 80 年代,C 开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言之一。

在微机上使用的 C 编译系统主要有 Microsoft C、Turbo C、Quick C、Borland C 等,它们不仅实现了 ANSI C 标准,而且还各自作了一些扩充,使之更加方便、完美。但是,它们相互之间略有差异,请读者参阅相应的手册,注意自己在上机时所使用的 C 编译系统的特点和规定。

如今,C++ 又成为了高级语言的主流,C++ 是面向对象的程序设计语言,但它的基础是 C 语言,且二者在很多方面是兼容的。因此,掌握了 C 语言,再去学习 C++,就会达到事半功倍的效果。

本书叙述以 ANSI C 为基础,C 语言上机实验时用的是 Turbo C 2.0,C++ 可使用 Borland C++ 3.1。

1.1.2 C 语言的特点

1. 与自然语言比较

自然语言是人类交流的重要工具,人类可以利用自然语言进行沟通和交流,共同完成生产和生活实践。人与计算机进行交流也是为了信息的交流,只是表达方式、规则等与人类交流有所不同。自然语言与 C 语言的比较如表 1.1 所示。

表 1.1 自然语言与 C 语言的比较

异同点	自然语言	C 语言
信息交流	交流双方地位平等 均有思维、推理能力	人机对话(采取命令方式) 计算机一般无思维、推理能力,具有计算 与逻辑判断能力

续表

异同点	自然语言	C语言
语法规则 句法规则	灵活(可省略、颠倒部分内容), 如:“走,上课去!” “上课去,走!” (上面两句话意思相同,且都省略了主语)	固定(一般不可省略、颠倒,必须按部就班), 如:scanf("%d%d",&a,&b); $x=a+b$; (上面两条C语句颠倒后可能得不到正确结果)
表达方式	多样	算法多样

2. 与其他程序设计语言比较

(1) C语言简洁、紧凑,使用方便、灵活。ANSI C一共只有32个关键字(见附录Ⅱ),如int、long、float、if、while、do等9种控制语句。程序书写自由,主要用小写字母表示,压缩了一切不必要的成分。

(2) 运算符丰富。共有34种运算符(见附录Ⅲ)。C语言把括号、赋值、逗号等都作为运算符处理,从而使C语言的运算类型极为丰富,可以方便地实现其他高级语言难以实现的功能。

(3) 数据结构类型丰富,具有现代语言的各种数据结构。C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。能实现各种复杂数据结构(如链表、树、栈等)的运算。尤其是指针类型数据,使用起来更为灵活、多样。

(4) 具有结构化的控制语句。用函数作为程序的基本单位,便于实现程序的模块化。C语言是良好的结构化语言,符合现代编程风格的要求。

(5) 语法限制不太严格,程序设计自由度大,如对数组下标越界不做检查;对变量的类型使用比较灵活,如整型数据与字符型数据可以通用。

(6) C语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此有人把它称为中级语言。

(7) 生成目标代码质量高,程序执行效率高,可达到汇编语言程序的80%。

(8) 与汇编语言相比,用C语言写的程序可移植性好。

C语言是理想的结构化语言,描述能力强,且现在操作系统课多结合UNIX讲解,而UNIX与C语言不可分,因此C语言已经成为被广泛使用的教学语言。C语言除了能用于教学外,还有广阔的应用领域,因此更有生命力。PASCAL和其他高级语言的设计目标是通过严格的语法定义和检查来保证程序的正确性,而C语言则是强调灵活性,使程序设计人员能有较大的自由度,以适应广泛的应用面。“限制”与“灵活”是一对矛盾,限制严格,就失去灵活性;而强调灵活,

就必然增加了出错的可能性。一个不熟练的程序设计人员,编写一个正确的 C 程序可能会比编写一个其他高级语言程序更难一些。也就是说,对使用 C 语言的人,要求对程序设计更熟练一些。总之,C 语言对程序员要求较高。但程序员使用 C 语言编写程序会感到限制少,灵活性大,功能强,可以编写出任何类型的程序。C 语言已不仅用来编写系统软件,也用来编写应用软件,因此学习和使用 C 语言的人已越来越多。

1.2 C 语言程序的基本结构

所谓程序,就是一系列遵循一定规则 and 思想并能正确完成指定工作的代码。使用 C 语言编写的程序称为 C 语言源程序(简称 C 语言程序或 C 程序)。下面介绍两个 C 程序,虽然其中涉及的内容还未介绍,但从中可了解到 C 程序的基本结构和书写格式。

例 1.1 求任意两个整数之和。

```
main()    /* 主函数 */  
{  
    int a,b,sum;    /* 定义整型变量 a,b,sum */  
    printf("Please input two integers:\n"); /* 输出提示信息,增加程序的交互性 */  
    scanf("%d%d",&a,&b); /* 输入两个整数,并赋给 a,b */  
    sum=a+b; /* 计算 sum */  
    printf("%d+%d=%d\n",a,b,sum); /* 输出结果 */  
}
```

程序运行情况如下:

```
Please input two integers:  
6 8 ↵ (↵表示空格,↵表示按 Enter 键)  
6+8=14
```

例 1.1 中只包含一个 main 函数。函数是具有特定功能的程序模块,由函数首部(起始行)和函数体构成。函数体由左花括弧“{”开始,到右花括弧“}”结束,其间可有声明部分和执行部分。声明部分是对函数中所要用到的变量等的定义或声明,而执行部分则是用来完成函数功能的语句段。此例函数首部为 main(),函数体内有五条语句,每条语句末尾用一分号作为语句结束标志。第一句为声明语句,表示定义了 a、b 和 sum 等整型(int)变量;其他四句为执行语句。第二句和第五句中用到了 printf 函数,它是 C 语言的一个库函数(系统已编

写好,用户可直接使用),其功能是把数据按照一定的格式输出到默认输出终端(显示器)。第二句中的 printf 函数输出一个字符串,“\n”是转义字符,表示换行;第五句中的 printf 函数括号内的部分是用逗号分隔开的参数列表。第一个参数为格式控制字符串,用一对双撇号(“”)括住,表示输出数据的格式;后边的参数为要输出的数据项(a、b、sum)。格式控制字符串中的三个“%d”表示后面的对应输出项以整型格式输出,其他字符为普通字符,原样输出。第三句中使用的 scanf 函数也是 C 语言的一个库函数,本语句的作用是从默认输入终端(键盘)输入两个整数并分别赋给变量 a 和 b。与 printf 函数类似,scanf 函数的第一个参数为输入数据的格式,也用双撇号括住;其后为变量的地址。有关这两个库函数的详细使用方法,将于第 3 章做进一步介绍。在程序中用“/*”和“*/”括起来的内容为注释部分,可以增加程序的可阅读性,程序不执行注释部分。

例 1.2 求两个整数中的较大者。

```
#include <stdio.h>    /* 编译预处理 */

main()                /* 主函数首部 */
{
    int x,y,z;          /* 定义整型变量 x、y、z */
    int max(int a,int b); /* 声明函数 max */
    printf("Please input two integers:\n"); /* 输出提示信息 */
    scanf("%d,%d",&x,&y); /* 输入 x,y 值 */
    z=max(x,y);         /* 调用 max 函数 */
    printf("The maximum number is %d.\n",z); /* 输出结果 */

    int max(int a,int b) /* max 函数首部 */
    {
        if(a>b) return a; /* 条件语句,实现选择结构 */
        else return b;    /* 把结果返回主调函数 */
    }
}
```

程序运行情况如下:

```
Please input two integers:
6,8
```

```
The maximum number is 8.
```

该程序中,第一行是一个编译预处理命令,表示把 include 后尖括号<>内的文件 stdio.h 直接包含到本程序中,成为本程序的一部分,因这些工作是在程序编译之前完成的,故得名编译预处理(详见第 8 章)。C 语言要求在使用函数

前必须声明,而文件 `stdio.h` 中有 `scanf` 和 `printf` 等标准库函数的函数声明,因此,在调用这些库函数之前,就应该包含 `stdio.h`。但是,由于这两个函数使用频率非常高,C 语言规定可省略对 `stdio.h` 的包含命令,如例 1.1。

例 1.2 程序共有两个函数:一个是主函数 `main`;一个是自定义的函数 `max`。在 `main` 函数中,定义了三个整型变量 `x`、`y`、`z`,输入的两个整数存于 `x` 和 `y`,这两个整数中的较大者存于 `z`。接下来对 `max` 函数进行声明。程序的第五行调用 `printf` 函数在显示器上输出提示信息,请操作人员输入两个整数。第六行调用 `scanf` 函数,接受键盘上输入的数并赋给变量 `x` 和 `y`。第七行调用自定义函数 `max`,求出 `x` 和 `y` 中的大者并赋给变量 `z`。第八行调用 `printf` 函数输出变量 `z` 的值,即 `x` 和 `y` 中的较大值。

在 `main` 函数之后便是用户自定义的 `max` 函数。`max` 函数的函数体中的 `if` 结构(详见第 4 章)用来完成求两个整数中的较大数,并将结果返回给调用函数。关于函数的详细内容将在第 7 章介绍。

由以上两个例子,可以看出:

- (1) C 程序可由一个或多个函数构成,函数是 C 程序的基本单位。
- (2) C 程序必须有一个且只能有一个 `main` 函数,即主函数。
- (3) 一个 C 程序总是从主函数开始执行,而不论主函数在整个程序中位置如何。主函数执行完了,整个程序也就执行完了。
- (4) C 程序中可以有编译预处理命令(如 `include` 命令)。
- (5) 每一个语句都必须以分号结尾。但编译预处理命令、函数首部(即函数的起始行)和花括号“`}`”之后不能加分号。
- (6) C 语言本身没有输入输出语句,输入输出功能是用输入输出库函数来实现的。
- (7) “`/ *`”和“`* /`”为注释符,二者之间的部分为注释。注释用来向用户提示或解释程序的意义,可出现在程序中的任何位置。编译时,不对注释作任何处理。在调试程序中对暂不使用的语句也可用注释符括起来,使编译器跳过不作处理,待调试结束后再去掉注释符。

另外,从书写清晰和便于阅读、理解、维护的角度出发,在书写程序时应尽量遵循以下规则:

- (1) 一个声明或一条语句占一行。当然 C 程序允许一行写多条语句,也允许一条语句写在多行上,且无需续行符。
- (2) 用 `{ }` 括起来的部分,通常表示程序的某一层结构。`{ }` 一般与该结构语句的第一个字母对齐,“`}`”最好单独占一行。
- (3) 低一层次的语句或声明可比高一层次的语句或声明缩进若干格后书

写。以便看起来更加清晰,增加程序的可读性。

在编程时应力求遵循这些规则,以养成良好的编程风格。

1.3 算法及其描述

1.3.1 算法的概念

1. 算法——程序的灵魂

一个程序应包括如下两种描述:

(1) 对数据的描述,在程序中要指定数据的类型和数据之间的组织形式,即数据结构。在C语言中,系统提供的数据结构是以数据类型的形式出现的。

(2) 对数据处理的描述,即计算机算法。广义地说,为解决一个问题而采取的方法和步骤,就称为“算法”,它是程序的灵魂。因此,著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式:

程序 = 数据结构 + 算法

实际上,一个程序除了数据结构和算法外,还必须使用一种计算机语言,并在必要的环境支持下,采用合适的程序设计方法来设计。因此,程序可以更完整地表达为:

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

这四个方面都是一个程序设计人员所应具备的知识。其中,算法是灵魂,数据结构是加工对象,语言是工具,编程需要采用合适的方法。本书旨在叙述怎样编写C语言程序,这里只介绍算法的初步知识。

从事各种工作和活动,都必须事先想好步骤,然后一步一步地进行,才能避免产生错乱。对同一个问题,又可以有不同的解决方法和步骤。方法有优劣之分,有的方法只需进行很少的步骤,而有些方法则需要较多的步骤。一般来说,希望采用简单的和步骤少的方法。因此,为了有效地解决问题,不仅需要保证算法的正确性,还要考虑算法的质量。

本书所关心的仅限于计算机算法,即计算机能执行的算法。

2. 算法的分类

计算机算法可分为两大类:数值算法和非数值算法。

数值运算的目的是求数值解,一般的数值运算都有现成的模型,可以运用数值分析方法进行解决,因此对数值运算的算法研究比较深入,算法也比较成熟。人们常常把这些成熟的算法汇编成册(写成程序形式),或者将这些程序存放在